DIVISION OF COMPUTER SCIENCE

The special requirements of multimedia systems: how to choose the most effective modelling notations.

Carol Britton
Sara Jones
Margaret Myers
Mitra Sharif

Technical Report No.244

April 1996

The special requirements of multimedia systems: how to choose the most effective modelling notations.

Carol Britton[1], Sara Jones[1], Margaret Myers[2], Mitra Sharif[1]

[1]School of Information Sciences,
University of Hertfordshire, Hatfield, AL10 9AB, UK.
Tel:+44-(0)1707-284321, Fax: +44-(0)1707-284303,
Email: C.Britton, S.Jones, M.Sharif@herts.ac.uk
[2]American International University in London,
Richmond College, Queens Road, Richmond. TW10 6JP
Tel: +44-(0)181-455-7921, Email: myersm@staff.richmond.ac.uk.

Classification:

Problem addressed:	2.6
Contribution:	G

Abstract

Although still in its infancy, multimedia system production is a growing and rapidly evolving industry. The nature of multimedia systems, incorporating text, graphics, animation, sound and video, means that the process of eliciting, specifying and validating requirements for such systems differs in many ways from the process of requirements capture in other software development. The literature on requirements engineering contains little in the way of either theoretical guidance or empirical case studies relating to the specification of requirements for interactive multimedia systems. Where advice does exist for multimedia systems, it is geared mainly to their design and production, rather than to the engineering of their requirements. No one has yet come forward with techniques or tools specifically designed to model requirements for multimedia systems; nor has any systematic evaluation of currently available approaches been carried out in this context.

In this paper we identify some of the distinctive features of multimedia systems that have a material effect on the requirements process. We then develop a list of criteria for assessing the suitability of notations for effective modelling of requirements for a multimedia system. We examine the extent to which different notations can be evaluated in relation to the criteria by the application of a formal measurement scheme. The use of the criteria and measurement scheme when selecting a notation to model requirements for a multimedia system will ensure that the choice of notation is soundly based and give confidence that the models produced will constitute a high quality specification of the system requirements. We believe that this rigorous approach to the selection of notations has proved useful in the area of modelling requirements for multimedia systems and that the same approach applied to systems in general will help to produce requirements specifications that are well-suited to the particular needs of design and development in different contexts.

1 Introduction

The nature of multimedia systems, incorporating text, graphics, animation, sound and video, means that the process of eliciting, specifying and validating requirements for such systems differs in many ways from the same process in traditional software development. The literature on requirements engineering contains little in the way of either theoretical guidance or empirical case studies relating to the specification of requirements for interactive multimedia systems. Where advice does exist for multimedia systems, it is geared mainly to their design and production, rather than to the engineering of their requirements. Currently the lack of guidance in the multimedia requirements process is not a problem, since multimedia systems have tended to be market-driven rather than customer specific, developed by relatively small teams of people, and used in areas which are neither safety- nor security-critical. However, with the demand for more complex multimedia systems in a wide range of areas, the prospect of a multimedia equivalent to the software crisis looks increasingly likely.

In this paper we describe criteria and a measurement scheme to support the formal evaluation of modelling notations for multimedia system requirements. Section two of the paper describes distinctive features of multimedia system development that affect the requirements process. In the third section we suggest a list of criteria for assessing the suitability of notations for modelling requirements for multimedia systems. Section four describes measures that can be applied to formalize the evaluation of notations in terms of these criteria. Finally, we illustrate the way in which the criteria and measures can be used in a small example of a multimedia system, and conclude that the same approach applied to modelling notations for systems in general will help to produce requirements specifications that are well-suited to the particular needs of design and development in different contexts.

The work described in this paper is part of the M3 (Modelling MultiMedia) project [Britton 1996], which is funded by the EPSRC.

2 Distinctive features of multimedia system development

The elicitation, specification and validation of requirements for interactive multimedia systems present the developer with a number of problems. Some of these problems are the same as those experienced by developers of traditional software systems, but others are more specific to the development of multimedia systems. In this section we briefly discuss some of the more distinctive features of multimedia projects that have a material effect on the multimedia requirements process. A more detailed account can be found in [Jones and Britton 1996].

2.1 Requirements for information presentation, rather than processing: Many multimedia information systems are better characterised as systems for information presentation, rather than information processing. The data stored in such systems does not change during the running of the system, so this means that many notations which have traditionally been used for modelling data processing requirements, such as data flow diagrams or entity life histories will be inappropriate. It also means that more emphasis must be placed on what have been called 'non-functional' requirements (relating, for example, to information content, usability and different media), than on input-output specifications of required functionality. Problems of this kind are not unique to the development of multimedia systems, but are likely to have more influence on the development process than in traditional systems.

2.2 Need for integrated modelling techniques: Multimedia systems will normally, by definition, integrate many different media, requirements for each of which might be modelled in a different way: for example, storyboards have traditionally been used to define sequences of video and animation, voice clips might typically be scripted in natural language, music might follow a score, and graphics and screen designs might be mapped out in free-form diagrams, whereas the connectivity of the underlying hypertext network might be modelled as a connection map, and the interactivity might be represented in terms of state transition diagrams. There is a need for a common framework within which requirements of all these kinds can be integrated. If particular kinds of requirements are omitted from this framework, there is a danger that the technology they relate to will be under-exploited: for example, if developers cannot see how to model the relations and connections between different screens, they may impose an overly simplistic structure on the system and thereby restrict its functionality.

This kind of problem is encountered in developing many kinds of system where requirements on different aspects or views of the system need to be integrated into a single specification, but it is particularly acute in the case of multimedia systems, owing to the large number and wide range of elements that need to be combined.

2.3 Confusion between requirements specification and detailed design: In multimedia system development, it may be especially difficult to distinguish between requirements and design. This can be problematic to the project manager as it makes it difficult to establish identifiable project milestones. Clients may have requirements that particular parts of the system should be built in very specific ways: for example, they may specify that particular video clips should be used, or a particular piece of music should be played. Precise specifications of videos or music may be viewed by some as being an essential part of the requirements specification, and by others as part of design.

2.4 The implications of using new technologies: Any development employing new or emerging technologies is vulnerable to problems associated with uncertainty. Lack of experience with the relevant technology means that members of the development team are likely both to over- and under-estimate the capabilities which the system might feasibly provide. Different members of the team start the project with different expectations, depending on their previous experience, and each member's expectations evolve during the course of the project, probably at a different rate from those of others. All these factors make communication both more necessary and at the same time more difficult.

The development of experience and understanding of new technologies through the course of a project can also lead to radical re-thinking of the domain, as new ways of doing things become apparent. Methods for requirements definition must therefore cater not only for the evolution of requirements, but also for evolution in the domain itself as new possibilities are discovered by the domain experts.

2.5 A fast-moving market: Related to the rapid development and evolution of multimedia technology is the fact that the market for multimedia systems is extremely fast-moving. More often than not, companies involved in multimedia development find themselves developing to market (rather than developing a bespoke system for a particular client) so that they can capture a particular market niche before any of their competitors. Developments obviously need to progress quickly in this context. We must therefore acknowledge the need for requirements practices and procedures which are flexible and efficient in delivering effective requirements with a minimum of cost, time and effort.

2.6 Disparity of user groups and development Teams: An important group of information systems are those which provide information to members of the general public in an easily accessible form. For these systems there is no easily identifiable 'user' from whom to elicit requirements. Educational information systems may be directed at a particular subset of the general population, but still have to cater for considerable differences in ethnic and social backgrounds, educational attainment, knowledge of the domain and experience or confidence with the use of computers. Methods for defining requirements for such systems need to be able to record and integrate requirements from representatives of many kinds of system users. Of course this is true for both mono- and multimedia system development.

A multimedia development project may also involve a large and multidisciplinary team of system developers. In multimedia system development, it is not possible to simply pass a software specification to a team of programmers - the development must, for example, integrate material provided by audio-visual technicians and specialists in computer graphics and animation. Writers, actors, directors, editors, composers and artists may also be involved, if less directly. Each of the team members will bring their own views and experiences to the project, which can enrich the development process, but also lead to problems of management and communication. Moreover, work on M3 [Britton 1996] has found that, in the case of many multimedia developments the main developer of the system does not have a computer science background. This means that layers of knowledge and experience in computer science which can be assumed in the developers of a traditional system do not exist in multimedia development.

In the next section we suggest criteria to support the choice of modelling notation for a multimedia system development in the light of the distinctive features of such systems identified above.

3 Criteria for modelling notations

Work on criteria for modelling notations for traditional computer systems has been carried out by authors from both the academic and industrial communities, including Farbey [1993], Garzotto [1995] and the STARTS guide [1987]. In this section we build on this work to suggest a set of criteria for notations to be used specifically for modelling multimedia system requirements. For descriptions of more generally applicable criteria, such as expressiveness, consistency, rigour and tool support, readers are referred to the authors mentioned above.

We should, however, point out that the importance given to each criterion will vary depending on the precise nature of the system and the main use which is to be made of the requirements specification. A requirements specification may be used in a variety of ways, for example:

(i) as a vehicle for communication and negotiation between client and developer and between members of the development team;
(ii) as a foundation for agreement and, if needed, a legal contract between developer and client;
(iii) as a basis for design and implementation of the system;
(iv) as documentation for future maintenance and modification of the system.

Each of these roles will influence the criteria used in the choice of modelling notation. As Green [1989] puts it : " A notation is never absolutely good, but good only in relation to certain tasks". For example, if the clients for the system are not computer literate and the requirements specification is to be used primarily as a vehicle for communication, ease of understanding of the notation for novice users will be a priority. On the other hand, if the development involves a complex combination of media and the main role of the requirements specification is as a basis for design and implementation, an important criterion will be the ability to model overall integration of different media.

3.1 Coverage (the ability of the notation to express different types of requirements): The concept of coverage has also been referred to as 'expressive adequacy' in [Duce 1988], it concerns the knowledge that can or cannot be represented by a particular notation.

For any system, we may assume the existence of a set of functional and non-functional requirements at the beginning of a project. These requirements may be written down, expressed verbally, or simply exist in the minds of the client or users. In most cases the requirements set will be a combination of all three of these and will include requirements from a variety of different sources. In addition, new requirements will emerge during development and become part of the set.

Different modelling notations allow different ideas to be expressed and are able to cover different requirements. For example, data flow diagrams enable the developer to represent the movement of data round the system, but not relationships between data items. This is covered by entity-relationship diagrams. Neither of these techniques is able to model the user interface. It is the responsibility of the system developer to ensure complete coverage of the requirements through the choice of notations used.

3.2 Usability and ease of understanding of the notation: The ISO Standard 9241 defines usability of computer systems as "the effectiveness, efficiency and satisfaction with which specified users can achieve specified goals in a particular environment by using the system". The usability of a notation will depend on who is regarded as the user of the notation (developer or client) and the nature of their individual expertise and experience. From the point of view of the developer, ease of writing and time to learn the notation are both important, but these will not be relevant for a client who only needs to be able to understand a model with the developer's help. Ease of understanding of the notation is, however, essential for both developer and client, since it is counter-productive for anyone trying to understand an idea to have to battle with the notation in which it is expressed. Ease of understanding is particularly important in a notation for modelling multimedia systems, because of the variety of background in members of the development team.

3.3 Ability of the notation to produce a model which exhibits a clear structure: It should be possible to decompose a model into ' brain-sized chunks' which mirror different concerns in the problem, and the interrelationship of the component parts should be clearly visible. The same criterion was recommended some time ago for programming languages in [Barron 1977]: "A desirable feature of a language is that it should assist the programmer to produce programs that are "nicely structured", in the sense that the structure of the program reflects directly the structure of the problem .... such programs will be easier to follow: they are less likely to contain errors, and the errors that they contain will be easier to find".

3.4 Compatibility with other modelling notations: Since there is no ' best' notation, most requirements specifications call for a combination of modelling notations. However, this will only be effective if the combination of notations used can produce good coverage of the requirements (see 3.1, above), and if it is possible to check the individual models for consistency with each other. Certain notations, such as data flow diagrams and data dictionaries are almost always used together, but other combinations may be equally productive. Compatibility of notations has a high priority in multimedia system development, since specialist team members from different backgrounds may wish to use their own modelling notations. The M3 [Britton 1996] project has found that virtually all multimedia developers use the storyboard notation, although this is not a technique that is traditionally used in computer systems development.

3.5 Ability to model two or more different types of media and to model the overall integration of different media: The need for a notation to represent a range of media and their overall integration is discussed in section 2.2.

Table 1, below, illustrates how the distinctive features of multimedia system developments outlined in section 2 relate to the criteria for choosing modelling notations for multimedia system requirements suggested above.

Distinctive features of multimedia system development	**Criteria for choosing modelling notations for multimedia system requirements**
Requirements for information presentation, rather than processing (2.1).	Coverage (particularly non-functional requirements) (3.1).
Need for integrated modelling techniques (2.2).	Compatibility with other modelling notations (3.4). Ability to model two or more different types of media and to model the overall integration of different media (3.5).
Confusion between requirements specification and detailed design (2.3).	Ability of the notation to produce a model which exhibits a clear structure (3.3).
Implications of using new technologies (2.4).	Coverage (the ability of the notation to express different types of requirements) (3.1). Usability and ease of understanding of the notation (3.2).
A fast-moving market (2.5).	Coverage (the ability of the notation to express different types of requirements) (3.1). Usability and ease of understanding of the notation (3.2). Ability to model two or more different types of media and to model the overall integration of different media (3.5).
Disparity of user groups and development teams (2.6).	Usability and ease of understanding of the notation (3.2).

Table 1: Distinctive features of multimedia system development and associated criteria for choosing modelling notations

4 Measuring notations in terms of the criteria

In order to ensure that evaluation of modelling notations against the criteria suggested in section 3 is as soundly based as possible, we need to formalize the process by providing explicit links between criteria and notations. However, the complexity of the criteria means that a single, direct measure applied to a notation (such as the number of distinct symbols that it contains) is unlikely on its own to provide meaningful information as to the extent to which the notation fulfils any particular criterion. To achieve a better picture of how well a notation satisfies a criterion, we apply a hierarchy of measurement techniques [Kaposi and Myers 1994] which is explained briefly below.

4.1 A brief introduction to measurement theory

4.1.1 Direct and indirect measures: Values of measures may be obtained directly by inspection, comparison with a standard, or instrumentation. Values not directly available may be deduced using domain theory, for example through knowing how area relates to linear measures of depth and breadth, and watts to volts and amps. Whether a value is obtained directly or indirectly depends only on the nature of the measure and the instrumentation available. The measurer must show how the real world object (the referent) is to be measured objectively, on what scale, in what dimensional units and to what order of accuracy.

4.1.2 Object oriented measures: A set of measures, all belonging or relating to the same referent or object are 'object oriented'. The referent is characterized by the combination of measures.

4.1.3 Utility measures: Where value judgements are to be made on a referent, then utility measures are defined in which the judge's subjective values are made explicit: the level of noise to be allowed at an airport being dependent on the population density, for instance. Utility measures stand at the pinnacle of the hierarchy of Figure 1. Below that level all measures are objective and verifiable to the required or stated degree of accuracy. Only at the final stage may the observer's views be incorporated. The measurer must state assumptions, justifying them where possible.

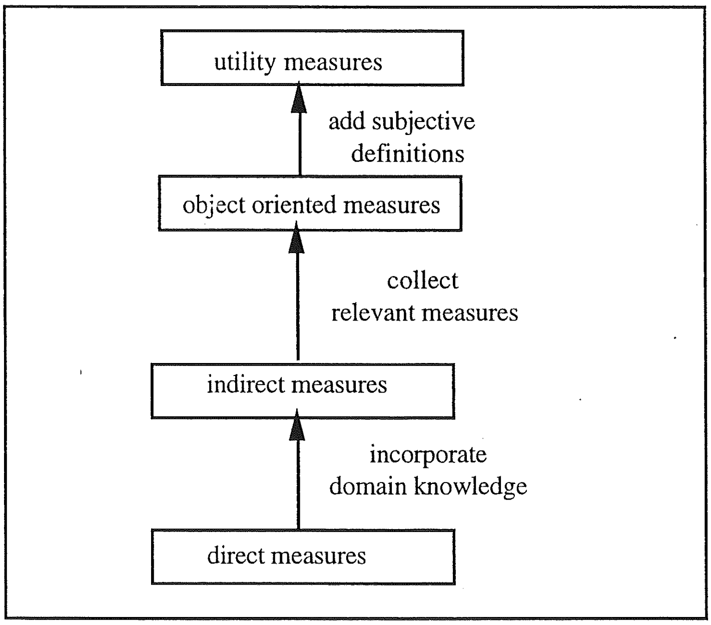
utility measures
add subjective definitions
object oriented measures
collect relevant measures
indirect measures
incorporate domain knowledge
direct measures

Figure 1: Hierarchy of measurement techniques

In the rest of this section we suggest possible measures relating to two of the criteria from section 3: ease of understanding (3.2) and coverage (3.1).

4.2 Measures to assess ease of understanding: Evaluation of the ease of understanding of a notation will always be ultimately subjective. There are so many factors which influence any given situation that any attempt at generalization will be of little practical use. What we have tried to do here is to identify certain properties of notations that may be used to support a subjective assessment and to illustrate how two of these properties may be included in a measurement scheme.

The properties of a notation that we have so far identified as contributing to the ease of understanding of a notation for a novice are outlined below.

4.2.1 Number of symbols in a notation: Notations with a large number of different symbols will be more difficult to understand for the novice user. This is a crude measure, since it does not take into account the amount of information contained in each individual symbol. It does, however, give some idea of the number of different symbols which the user may encounter in a model.

The number of symbols in a notation can be measured by the number of identifiable separate entries in a key or glossary as shown in Table 2, below. The table illustrates the huge difference between the number of symbols that a novice may encounter in a model using the storyboard notation, or one of the structured techniques, and a model built using a formal specification language, such as Z.

Notation	Number of entries in key or glossary	References and comments
DFD	4	Fertuck 1992
Structure Charts	5	Fertuck 1992
ERD (Chen)	5	Fertuck 1992
State Transition diagrams	4	Fertuck 1992
Z	76	Spivey 1987
1st order Predicate logic	12	Turski and Maibaum 1987
Data Dictionary	8	7 shown in Yourdon 1989 + legal character string
Storyboard	2	Box and line
Natural Language	8	Common parts of speech (adjective, adverb, article, conjunction, noun, preposition, pronoun, verb)

Table 2: Number of symbols in notation

4.2.2 Discriminability of symbols [Green 1980]: This refers to the ease with which different symbols in the notation can be distinguished from each other. As an example, we can consider the Z symbols for partial function and maplet, shown below:

⇸	↦
partial function	maplet

These are difficult to distinguish from each other and so are a potential cause of confusion for anyone trying to understand a specification written in Z.

4.2.3 Perceptual or symbolic [Fitter and Green 1981]: In general, perceptual representations are those in which we perceive meaning directly without having to reason about them; they may appeal to any of the senses, for example use of colour in electricity cables, auditory tones in the telephone system and diagrams in various contexts. Only the last of these counts as a notation and need concern us here.

In practice both perceptual and symbolic notations frequently mix in the same model. Graphs are diagrammatic but often contain labels (symbols) for the various nodes. Symbolic notation is frequently laid out to show certain aspects perceptually: Z's use of schemas, formatting of program code and introduction of 'white space' to aid comprehension are examples of this.

Diagrammatic and symbolic notations each have advantages and disadvantages. Models in diagrammatic notation often embody ideas such as connectedness (as used in data flow diagrams) and inclusiveness (as used in Venn diagrams), which most people understand. However, without elaborate 'extras' the amount of information they can contain is restricted. Models in symbolic notations are able to hold much more information and can be tested for such properties as consistency and completeness. However, they are mostly beyond the understanding of people not specially trained in the individual methods.

Table 3, below, summarizes the perceptual/symbolic classification for the modelling notations shown in Table 2, with a further subdivision into diagrammatic-connected and diagrammatic-inclusive attributes.

Diagrammatic notations are generally believed to be more accessible than symbolic for a novice user: "A picture is worth a thousand words". A preference for connected or inclusive notations will depend on the individual user and the situation. This is discussed further in section 4.2.4 on motivation.

Notation	perceptual		symbolic	comments
	diagrammatic -connected	diagrammatic -inclusive		
DFD	yes	no	no	
Structure Charts	yes	no	no	
ERD (Chen)	yes	no	no	
State Transition diagrams	yes	no	no	
Z	no	yes	yes	symbols within frames
Predicate logic	no	no	yes	
Data Dictionary	no	no	yes	
Storyboard	yes	no	yes	much used in industry
Natural Language	no	no	yes	for comparison

Table 3: Perceptual/symbolic classification

4.2.4 Motivation [Sampson 1985]. A notation may be considered to be motivated if there exists a natural relationship between the elements of the notation and objects or ideas that they represent. Many of the characters in Chinese script are motivated, for example:

木 and 林

meaning tree meaning forest

In an arbitrary (non-motivated) notation there is no natural relationship between the object and the representation; the symbol +, for example, bears no relation to the notion of conjunction.

Modelling notations for traditional software systems tend to be arbitrary, although some diagrammatic notations do include motivated symbols, such as a line representing a link in entity relationship diagrams, and an arrow signifying direction in data flow diagrams. Connected and inclusive diagrams (see section 4.2.3, above) may each be considered as motivated in certain situations. For example, a diagrammatic-inclusive notation can represent the positions of different elements on a screen, but for specifying possible routes through a system a diagrammatic-connected notation will be more effective.

Neither motivation nor discriminability (section 4.2.2) can currently be measured other than by subjective judgement, however, it is useful to be aware of both characteristics when evaluating notations in terms of ease of understanding.

4.3 Measures to assess coverage: Object oriented measurement (see section 4.1.2) may be applied to a notation in order to assess the ability of the notation to express different types of requirement.

We identify eight facets of multimedia systems which may be represented using a modelling notation and examine the ability of a number of notations to represent the various facets. The results are shown in Table 4, below. The number of affirmatives in each row of the table may be taken as a measure of the coverage ability of the notation. Results from the measurement cannot be absolute, but the measures can show comparisons of coverage between different notations.

From a study of Table 4 we can see that the notations which provide the best coverage of different types of requirements are natural language and storyboard (both of which we might have expected) and state transition diagrams (which we might not have anticipated.)

Notation	model data	model processes	model process sequence	model process concurr-ency	model the user	model the inter-face	model perform-ance	model different media
DFD	no	yes	no	no	no	no	no	no
Structure Charts	no	yes	yes	no	no	no	no	no
ERD (Chen)	yes	no	no	no	no	no	no	no
State Transition Diagrams	as state	yes	yes	yes	user actions	effect of actions	no	no
Z	yes	yes	no	no	no	no	no	no
Predicate logic	yes	no	no	no	no	no	no	no
Data Dictionary	yes	no	no	no	no	no	no	no
Story-board	yes	yes	yes	yes	no	yes	yes	yes
Natural Language	yes	yes	yes	yes	yes	yes	yes	yes

Table 4: Facets of multimedia systems represented by notations

5 Application of the criteria and measurement scheme: an example

As an example of the way in which the criteria and measurement scheme may be used to choose notations for modelling multimedia system requirements, we consider below the recent development of a multimedia system to encourage rejection of smoking in primary school children [Jones et al. 1995]. The main features of the development were as follows:

- the principal aim of the system was to present information that would affect the attitudes and behaviour of the child users with regard to smoking. The system would not carry out any processing of data;
- the system was not intended to be particularly sophisticated and would not involve complex media combinations and interactions;
- the domain experts were unfamiliar with multimedia systems and they had little experience of computers;
- there was a large disparity in the intended user group which included primary school children, school nurses and teachers;
- the development team consisted of four people, three of whom were experienced in computer system development. The fourth member of the team was an audio-visual technician.

In this context, communication with domain experts, potential users and the audio-visual technician was a significant issue. None of these people had experience of computer system development or notations commonly used in software engineering. Given these factors, ease of understanding was an important criterion in the choice of modelling notation.

The properties of a notation which relate to ease of understanding are:

- number of symbols in the notation (4.2.1);
- discriminability of symbols (4.2.2);
- perceptual v. symbolic representations (4.2.3);
- motivation of symbols in the notation (4.2.4).

In general we might expect that, in order to be easily understood by novices, a notation will be diagrammatic, with a low number of symbols, high discriminability and high motivation.

For the example of the anti-smoking system , we might decide to use storyboards to model a sequence of screens representing information about health, since storyboards have a low number of symbols, with high discriminability, use diagrammatic representations and are well motivated in this context. However, to specify requirements relating to possible routes that the user might follow through the system, state transition diagrams might be more appropriate, as the number of symbols used is still low, discriminability is again reasonable and the motivation is stronger than that of storyboards in this context. This is in line with current industrial practice, confirming our confidence in the criteria and measurement scheme.

6 Conclusion

Modelling plays a crucial part in the requirements process. It is important not only because of deliverables such as the requirements specification, documentation and prototypes, but also because it is through the process of modelling that the developer comes to understand the problem and to visualize a solution.

The importance of using an effective notation for modelling has long been recognized. For example, we only need to think of performing long multiplication and division using Roman numerals to realize the enormous benefits of the Arabic numeric system. Our problem in developing systems using new and rapidly evolving technology, such as multimedia, is how to have confidence in our choice of notation, since we have no way of knowing from experience which notation is the most effective.

In this paper we have suggested a set of criteria to assist the choice of notations for modelling multimedia system requirements. The criteria are supported by a measurement scheme which helps to formalize the process of evaluating notations. Much work remains to be done in this area, in particular investigation into further possible measures for criteria, such as those suggested above. The criteria and measures which have so far been identified need to be evaluated by using particular notations in practice. The next stage of the M3 project is to apply notations chosen using the criteria and measurement scheme to a number of multimedia case studies. Subjective evaluations of notations, for example in the areas of discriminability and motivation, should be validated by experiment. This is beyond the scope of the M3 project, but is nonetheless an important part of the work to be done.

In conclusion, we believe that this rigorous approach to the selection of notations has proved useful in the area of modelling requirements for multimedia systems. in addition, we suggest that the same approach applied to systems in general might help to produce requirements specifications that are well-suited to the particular needs of design and development in a variety of different contexts.

References

Barron, D.W. (1977) *An Introduction to the study of programming languages.* Cambridge University Press

Britton, C.E. (1996) *The Evaluation of tools and techniques suitable for modelling interactive multimedia systems used in teaching and training*, EPSRC grant reference GR/K62439

Duce,D. and Ringland,G. (Eds.) (1988) *Approaches to Knowledge representation : An Introduction* Research Studies Press Ltd., Letchworth, Herts.

Farbey, B. (1993)*Software quality metrics: considerations about requirements and requirement specifications* in Software Engineering: A European Perspective Thayer, R. and McGetterick, A. (Eds.), IEEE C.S.Press

Fertuck L. (1992) *Systems Analysis and Design*, WCB.

Fitter, M.J. and Green,T.R.G. (1981) *When do diagrams make good computer languages ?* from Alty,J. and Coombs,M. (Eds.) *Computing Skills and the User Interface* Academic Press.

Garzotto,F., Mainetti,L. Paolini,P. (1995) Hypermedia Design, Analysis and Evaluation Issues. Communications of the ACM Vol 38, No.8

Green T R G. (1989) *Cognitive Dimensions of Notations, in People and Computers,* Eds: Sutcliffe and Macaulay, COP.

Green,T.R.G. *Programming as a Cognitive Activity* in Smith,H.T. and Green,T.R.G. (Eds.) (1980) Human Interaction with Computers. Academic Press

Jones, S. and Britton, C. (1996) *Early Elicitation and Definition of Requirements for an Interactive Multimedia Information System* Proceedings of the Second International Conference on Requirements Engineering, IEEE Computer Society Press.

Jones, S., Britton, C., Knight, D. and Whiteley, N. (1995) *Requirements for a Multimedia Health Information System*, University of Hertfordshire, School of Information Sciences, Technical Report No. 224.

Kaposi A, Myers M,. (1994) *Systems Models and Measures,* Springer Verlag

Sampson, G. (1985) *Writing Systems.* Hutchinson.

Spivey J M. (1987) *The Z Notation A Reference Manual*, Prentice Hall

Department for Trade and Industry and National Computing Centre (1987) *The STARTS Guide,* second edition, volume 1, NCC Publications

Turski, W. and Maibaum, T. (1987) *The specification of computer programs* Addison-Wesley

Yourdon E. (1989) *Modern Structure Analysis,* Prentice Hall.